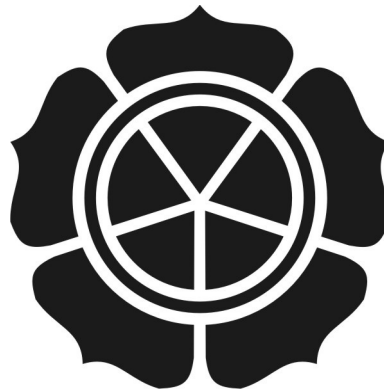


**PEMBUATAN APLIKASI MOBILE UNTUK MENGETAHUI JALUR BUS  
TRANS JOGJA**

**NASKAH PUBLIKASI**



diajukan oleh

**Muhammad Iqbal Basuki**

**05.11.0786**

kepada

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**AMIKOM**

**YOGYAKARTA**

**2010**

**NASKAH PUBLIKASI**

**PEMBUATAN APLIKASI MOBILE UNTUK MENGETAHUI JALUR BUS  
TRANS JOGJA**

disusun oleh

**Muhammad Iqbal Basuki**

**05.11.0786**

**Dosen Pembimbing,**

**Armadyah Amborowati, S.Kom., M.Eng.**

**NIK. 190302063**

**Tanggal 9 Agustus 2010**

**Ketua Jurusan**

**Teknik Informatika**



**Ir. Abas Ali Pangera, M.Kom.**

**NIK. 190302010**

# **THE MAKING OF MOBILE APPLICATION FOR KNOWING TRANS JOGJA BUS ROUTES**

## **PEMBUATAN APLIKASI MOBILE UNTUK MENGETAHUI JALUR BUS TRANS JOGJA**

Muhammad Iqbal Basuki  
Jurusan Teknik Informatika  
STMIK AMIKOM YOGYAKARTA

### ***ABSTRACT***

*The development of mobile phone technology is very rapid. Mobile phone has turned into one multi-function devices, one of which is often used now is to run mobile applications as a medium for accessing and processing information. The development of mobile applications supported by the development of programming languages, one of the many mobile programming language used is J2ME (Java 2 Micro Edition). By applying these technological advances, expected the world of transportation would be far more advanced in information systems accessible to users.*

*Trans Jogja is one part of the program implementation of Bus Rapid Transit (BRT), announced the Department of Transportation. This system began operation in early March 2008 by the Department of Transportation, the Provincial Government of Yogyakarta Special Region (DISHUB DIY). At a scheduled bus from 5:30 a.m. to 21:30 pm, Trans Jogja has six bus lines that stop at specific shelters. Media information on existing routes and bus stops are now just a map pinned to the Trans Jogja shelters as well as some brochures that were distributed directly to the public, but not all people have a map of the Trans Jogja because Dishub DIY print only those with a limited number of maps.*

*In this thesis the author attempts to develop a mobile application technology by creating an application that can be used to determine the bus lane Trans Jogja by utilizing the Java 2 Micro Edtion technology. With this application, users can find the desired bus lines and to find out a route that crossed by the routes.*

**Key words:** *Mobile Application, Java 2 Micro Edition Technology, Trans Jogja*

## **1. Pendahuluan**

Perkembangan teknologi telepon seluler sangatlah pesat. Telepon seluler telah berubah menjadi salah satu perangkat multi fungsi, salah satunya yang sering digunakan sekarang ini adalah untuk menjalankan aplikasi – aplikasi *mobile* sebagai media untuk mengakses dan mengolah informasi. Perkembangan aplikasi *mobile* didukung dengan semakin berkembangnya bahasa pemrograman, salah satu bahasa pemrograman *mobile* yang banyak digunakan adalah J2ME (*Java 2 Micro Edition*). Dengan menerapkan kemajuan teknologi ini, diharapkan dunia transportasi akan jauh lebih maju dalam sistem informasi yang dapat diakses pengguna.

Trans Jogja merupakan salah satu bagian dari program penerapan *Bus Rapid Transit* (BRT) yang dicanangkan Departemen Perhubungan. Sistem ini mulai dioperasikan pada awal bulan Maret 2008 oleh Dinas Perhubungan, Pemerintah Provinsi Daerah Istimewa Yogyakarta (DISHUB DIY). Bus terjadwal dari pukul 05.30 - 21.30 WIB, Trans Jogja memiliki 6 jalur bus yang berhenti di halte-halte khusus. Media informasi jalur dan halte yang ada sekarang hanya berupa peta yang ditempelkan pada halte-halte Trans Jogja serta beberapa brosur yang dibagikan langsung kepada masyarakat umum tapi tidak semua masyarakat memiliki peta Trans Jogja dikarenakan DISHUB DIY hanya mencetak peta tersebut dengan jumlah terbatas.

## **2. Landasan Teori**

### **2.1 Definisi Sistem, Informasi, Sistem Informasi**

#### **2.1.1 Definisi Sistem**

Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

Sistem juga merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan.

Suatu sistem yang baik harus mempunyai tujuan dan sasaran yang tepat karena hal ini akan sangat menentukan dalam mendefinisikan masukan yang dibutuhkan sistem dan juga keluaran yang dihasilkan. Sistem selalu berorientasi pada tujuan.

### **2.1.2 Definisi Informasi**

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa informasi maka organisasi tersebut tidak akan dapat berjalan dan tidak bisa beroperasi. Sumber informasi adalah data.

Kualitas informasi bergantung pada 3 hal yaitu keakuratan, tepat waktu, dan relevan.

### **2.1.3 Definisi Sistem Informasi**

Sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut. Selain itu data juga memegang peranan penting dalam sistem informasi. Data yang akan dimasukkan ke dalam sistem informasi dapat berupa formulir-formulir, prosedur-prosedur, dan bentuk data lainnya.

## **2.2 Karakteristik Sistem Informasi**

Untuk mendukung lancarnya suatu sistem informasi dibutuhkan beberapa komponen yang fungsinya sangat vital di dalam sistem informasi. Komponen-komponen yang terdapat pada sistem informasi antara lain meliputi:

### **1. *Input***

*Input* di sini adalah semua data yang dimasukkan ke dalam sistem informasi. Dalam hal ini yang termasuk di dalam *input* adalah dokumen-dokumen, formulir-formulir, dan *file-file*. Dokumen-dokumen tersebut dikumpulkan dan dikonfirmasi ke suatu bentuk sehingga dapat diterima oleh pengolah yang meliputi Pencatatan, Penyimpanan, Pengujian, dan Pengkodean.

### **2. Proses**

Proses merupakan kumpulan prosedur yang akan memanipulasi *input* yang kemudian akan disimpan dalam bagian basis data dan seterusnya akan diolah menjadi suatu *output* yang akan digunakan oleh si penerima.

### **3. *Output***

*Output* merupakan semua keluaran atau hasil dari model yang sudah diolah menjadi suatu informasi yang berguna dan dapat dipakai penerima. Komponen ini berhubungan langsung dengan pemakai dan merupakan tujuan akhir dari pembuatan sistem informasi.

#### 4. Teknologi

Teknologi disini merupakan bagian yang berfungsi untuk memasukkan *input*, mengolah *input* dan menghasilkan keluaran. Hal ini meliputi *hardware* misalnya *keyboard* dan *mouse*, *software* misalnya program untuk mengolah data, dan *brainware* misalnya analis sistem dan *programmer*.

#### 5. Basis Data

Basis data merupakan kumpulan-kumpulan data yang saling berhubungan satu dengan yang lain yang disimpan dalam perangkat keras komputer dan akan diolah menjadi perangkat lunak. Basis data sendiri merupakan kumpulan *file-file* yang mempunyai kaitan antara satu *file* dengan *file* yang lain sehingga membentuk satu bangunan data.

#### 6. Kendali

Kendali dalam hal ini merupakan semua tindakan yang diambil untuk menjaga sistem informasi tersebut agar bisa berjalan lancar dan tidak ada gangguan.

### 2.3 Aplikasi Mobile

Aplikasi *Mobile* adalah sebuah aplikasi yang memungkinkan Anda melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler atau *Handphone*. Dengan menggunakan aplikasi *mobile*, Anda dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing* dan lain sebagainya.

Pemanfaatan aplikasi *mobile* untuk hiburan paling banyak digemari oleh hampir 70% pengguna telepon seluler, karena dengan memanfaatkan adanya fitur game, music player, sampai video player membuat kita menjadi semakin mudah menikmati hiburan kapan saja dan dimanapun.

### 2.4 Unified Modeling Language (UML)

UML adalah suatu bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. UML memiliki diagram grafis sebagai berikut:

- a. *Use Case Diagram*
- b. *Class Diagram*
- c. *Activity Diagram*

Tujuan utama UML, antara lain :

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan

mudah dan dimengerti secara umum.

2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam permodelan.

## **2.5 Perangkat Lunak Yang Digunakan**

### **2.5.1 Netbeans 6.5**

NetBeans IDE adalah IDE *open source* yang ditulis sepenuhnya dengan bahasa pemrograman *Java* menggunakan *platform* NetBeans. Netbeans IDE mendukung pengembangan semua tipe aplikasi *Java* (J2SE, *web*, EJB, dan aplikasi *mobile*). Fitur lainnya adalah sistem proyek berbasis *Ant*, kontrol versi, dan *refactoring*.

## **2.6 Bahasa Pemrograman Yang Digunakan**

### **2.6.2 Java 2 Micro Edition**

J2ME merupakan teknologi *Java* yang digunakan untuk aplikasi *mobile device* yang memiliki *memory* terbatas. Misalnya telepon seluler, PDA (*Personal Digital Assistant*) dan sejenisnya.

## **3. Analisis Dan Perancangan Sistem**

### **3.1 Tinjauan Umum**

Trans Jogja adalah sebuah sistem transportasi bus cepat, murah dan ber-AC di seputar Kota Yogyakarta, Indonesia. Trans Jogja merupakan salah satu bagian dari program penerapan *Bus Rapid Transit* (BRT) yang dicanangkan Departemen Perhubungan. Sistem ini mulai dioperasikan pada awal bulan Maret 2008 oleh Dinas Perhubungan, Pemerintah Provinsi DIY. Motto pelayanannya adalah "Aman, Nyaman, Andal, Terjangkau, dan Ramah lingkungan".

Trans Jogja merupakan sebuah upaya Pemerintah Provinsi DIY untuk meningkatkan pelayanan publik khususnya pada sektor transportasi darat di kawasan perkotaan DIY dengan berbasis bus mengganti sistem setoran menjadi sistem pembelian pelayanan. Bus terjadwal dari pukul 05.30 - 21.30 WIB, Trans Jogja memiliki 6 jalur bus yang berhenti di halte-halte khusus.

## **3.2 Analisis**

### **3.2.1 Analisis Sistem**

Analisis sistem dapat didefinisikan sebagai penguraian dari sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Analisis sistem merupakan salah satu tahapan pada pengembangan sistem. Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan di dalam tahap ini juga akan menyebabkan kesalahan di tahap selanjutnya. Tahap analisis dilakukan setelah tahap perancangan sistem dan sebelum tahap desain sistem.

Berdasarkan penelitian, maka analisis yang tepat untuk mengetahui sistem adalah analisis SWOT, yaitu metode perencanaan strategis yang digunakan untuk mengevaluasi kekuatan, kelemahan, peluang, dan ancaman dalam suatu proyek proses ini melibatkan penentuan tujuan yang spesifik dari spekulasi proyek dan mengidentifikasi faktor internal dan eksternal, yang mendukung dan yang tidak dalam mencapai tujuan tersebut.

### **3.2.2 Analisis Kebutuhan Sistem**

Analisis kebutuhan sistem sangat dibutuhkan dalam mendukung kinerja sistem, apakah sistem yang dibuat telah sesuai dengan kebutuhan atau belum karena kebutuhan sistem akan mendukung tercapainya tujuan suatu instansi atau perusahaan. Analisis dibagi menjadi 2 yaitu Fungsional dan non Fungsional.

### **3.2.3 Analisis Kelayakan Sistem**

Analisis ini digunakan untuk menentukan kemungkinan apakah mengembangkan proyek sistem ini dapat diteruskan atau tidak. Ini dilakukan agar sistem baru tersebut dapat benar-benar bermanfaat. Analisis ini dilihat dari segi kelayakan teknologi, hukum, dan operasional.



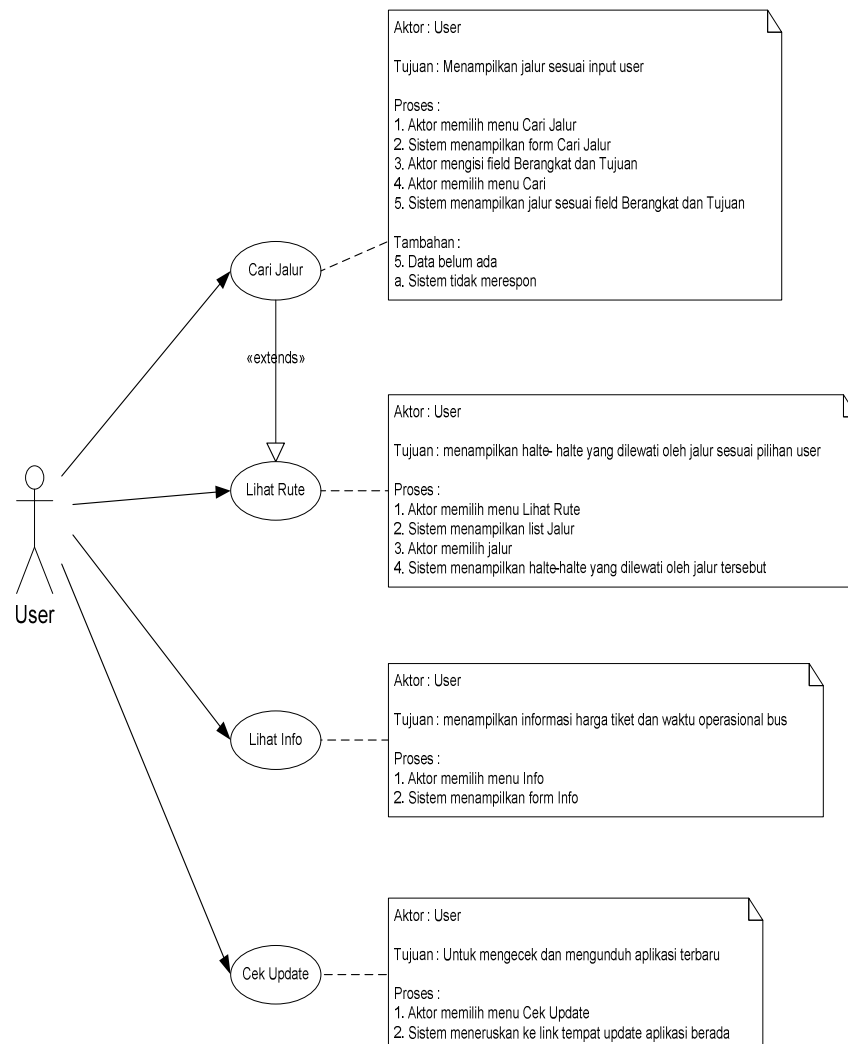
### 3.3 Perancangan Sistem

#### 3.3.1 Perancangan Proses

Perancangan proses disajikan dalam bentuk *logic model* dengan menggunakan *tools* UML (*Unified Modeling Language*). UML digunakan untuk menggambarkan urutan proses yang terjadi pada sistem berdasarkan *actor* dan fungsionalitas masing-masing *actor* yang terlibat dalam sistem ini. *Actor* menggambarkan pengguna aplikasi ini (*user*).

##### 1. Use Case Diagram

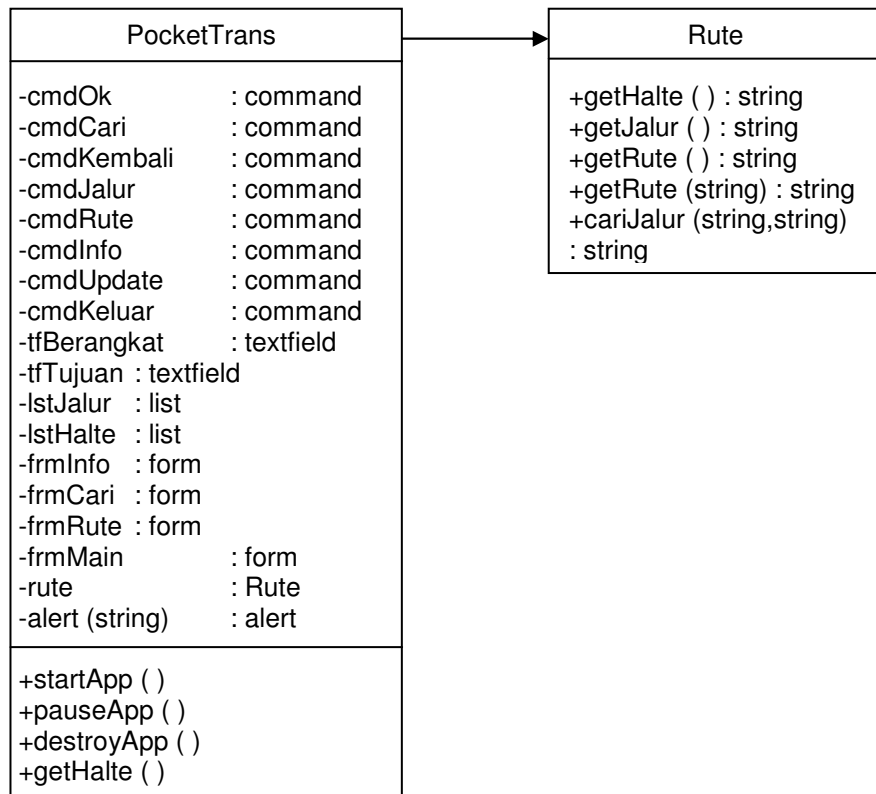
Perancangan proses terjadi pada sistem ini menggunakan *use case diagram* yang menekankan pada apa yang diperbuat oleh sistem tersebut. Pada *use case diagram* akan digambarkan sebuah interaksi antara *actor* dengan sistem.



Gambar 3.1 Use case fungsional actor user

## 2. Class Diagram

*Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

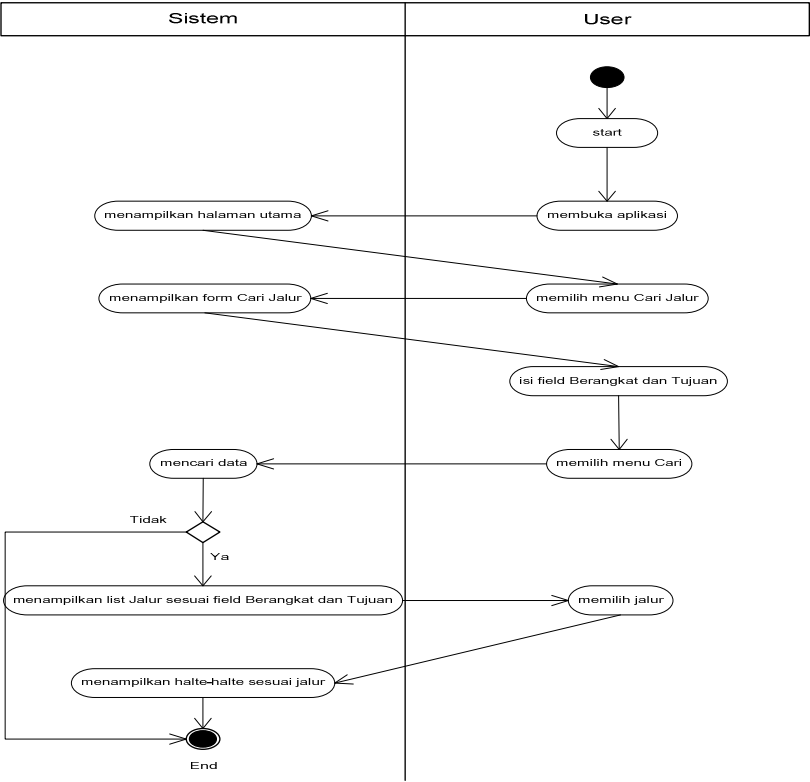


Gambar 3.2 Class diagram aplikasi mobile trans jogja

## 3. Activity Diagram

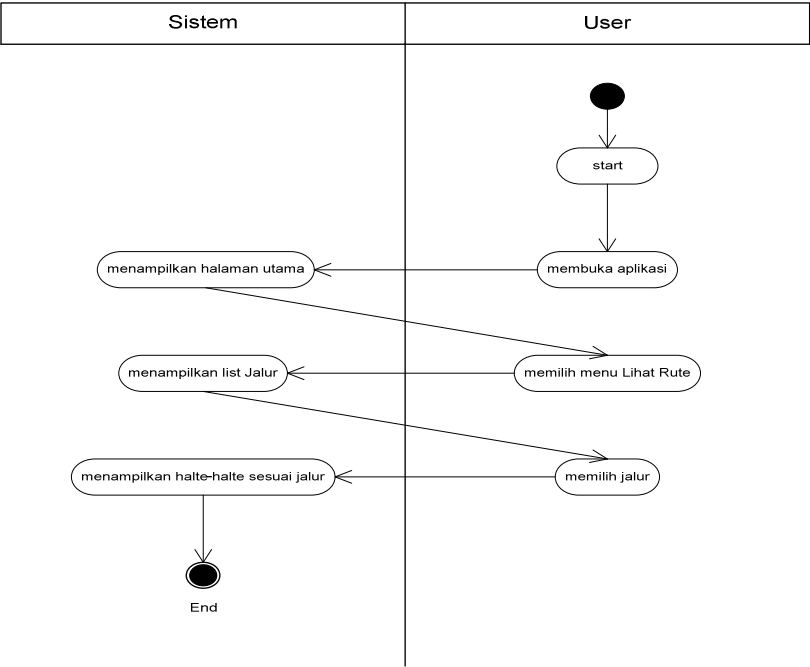
*Activity diagram* memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini mirip dengan sebuah *flowchart* karena kita dapat memodelkan sebuah alur kerja dari suatu aktivitas ke aktivitas lainnya. Membantu memodelkan sebuah proses dalam membantu kita memahami proses secara keseluruhan. Berikut merupakan *activity diagram* dari kegiatan di sistem ini.

a. Activity Diagram Cari Jalur



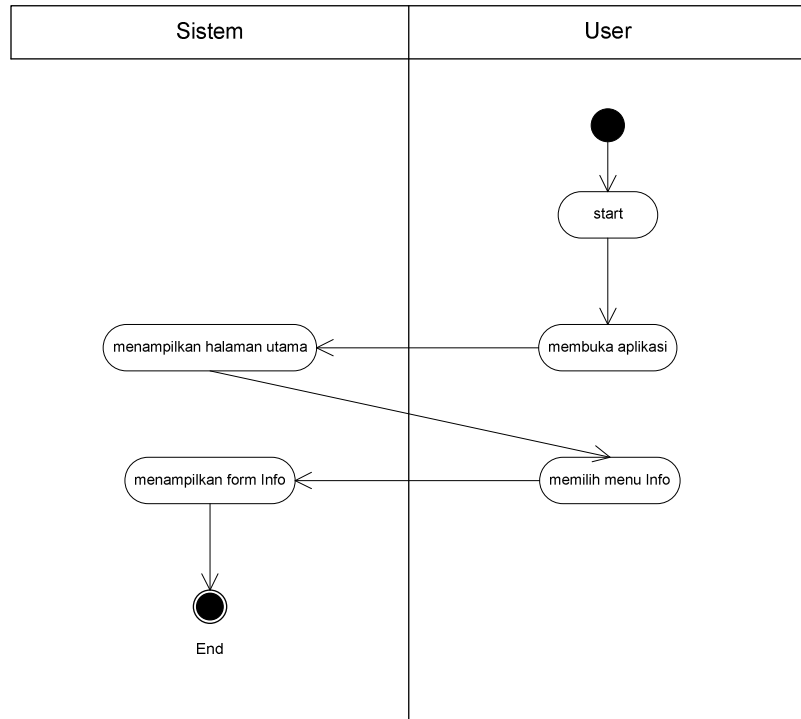
Gambar 3.3 Activity diagram cari jalur

b. Activity Diaram Lihat Rute



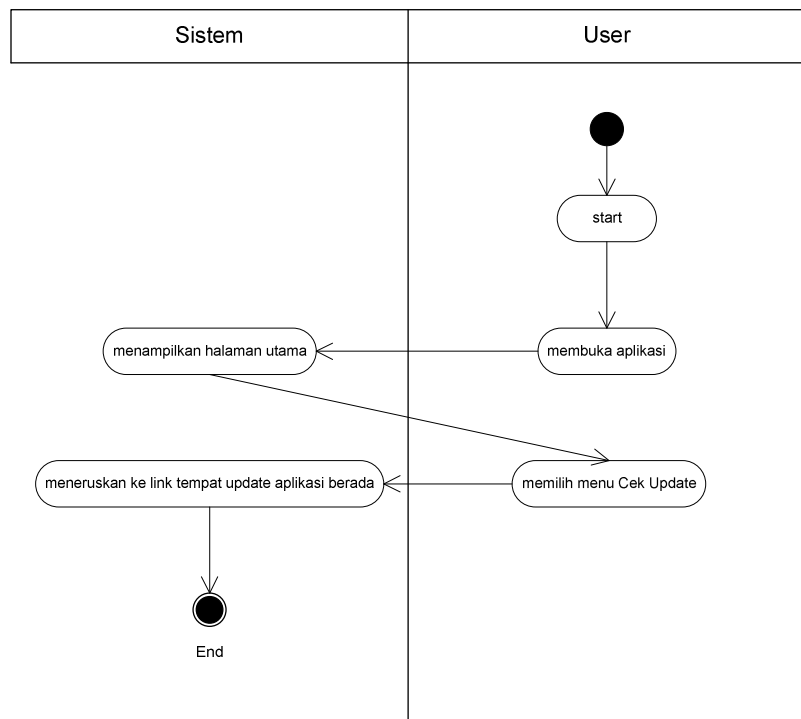
Gambar 3.4 *Activity diagram* lihat rute

c. *Activity Diagram* Lihat Info



Gambar 3.5 *Activity diagram* lihat info

d. *Activity Diagram* Cek Update



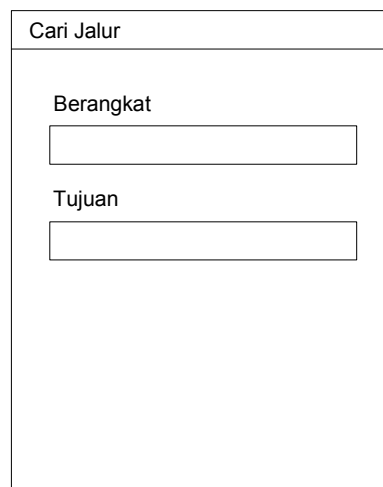
Gambar 3.6 *Activity diagram cek update*

### 3.3.2 Perancangan Aturan Jalur Dan Halte Bus Trans Jogja

Aturan sistem ini menggunakan file teks yang sudah satu paket dalam *software* aplikasinya. *File – file* teks tersebut digunakan untuk menyimpan nama-nama halte Trans Jogja dan rute tiap–tiap jalur.

### 3.3.3 Perancangan Antarmuka (*Interface*)

Perancangan *interface* dibagi menjadi beberapa tampilan yang berguna untuk *input* maupun *output* informasi di sistem ini. Sebagai contoh adalah *rancangan interface* form cari jalur.



The image shows a rectangular form titled "Cari Jalur" at the top. Below the title, there are two input fields. The first field is labeled "Berangkat" and the second field is labeled "Tujuan". Both fields are empty text boxes.

Gambar 3.7 Rancangan *interface form* cari jalur

## 4. Implementasi Dan Pembahasan Sistem

### 4.1 Implementasi

#### 4.1.1 Implementasi Aturan Jalur Dan Halte Bus Trans Jogja

Tahapan implementasi dilakukan dengan menerapkan hasil rancangan pada bagian sebelumnya agar aplikasi dapat dijalankan. Aturan untuk aplikasi ini terdiri dari beberapa *file* teks yang dapat dibuat dengan editor teks biasa.

#### 4.1.2 Implementasi Antarmuka (*Interface*)

##### 1. *Form* Halaman Utama

Halaman Utama terdiri terdapat 5 menu pilihan, yaitu cari jalur, lihat rute, cek *update*, info, dan keluar.



Gambar 4.1 *Interface form* halaman utama

## 2. *Form* Cari Jalur

*Form* cari jalur terdiri dari 2 *textfield* untuk meng-*input*-kan halte berangkat dan tujuan sesuai pencarian *user*.

Gambar 4.2 *Interface form* cari jalur

### 3. List Halte

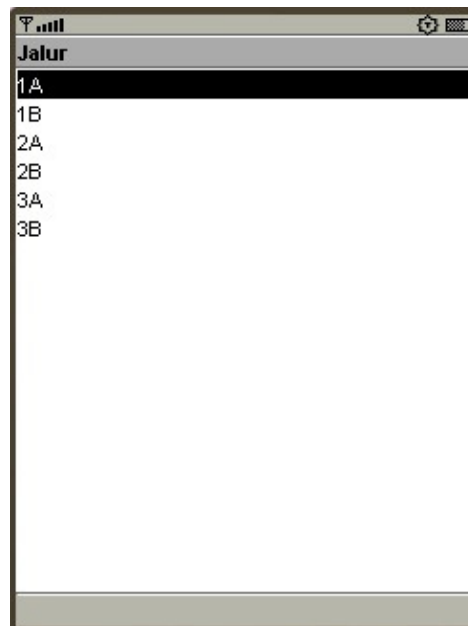
List halte menampilkan daftar halte yang mirip dengan *input* pada *form* cari jalur.



Gambar 4.3 Interface list halte

### 4. List Jalur

List jalur menampilkan jalur bus secara keseluruhan atau yang sesuai dengan pencarian pada *form* cari jalur.



Gambar 4.4 Interface list jalur

##### 5. *Form Rute*

*Form rute* menampilkan rute atau halte-halte yang dilalui oleh suatu jalur.



Gambar 4.5 *Interface form rute*

##### 6. *Form Info*

*Form Info* menampilkan informasi tarif, jadwal operasional bus, dan beberapa info singkat seputar Trans Jogja.



Gambar 4.6 *Interface form info*



## 4.2 Uji Coba Program

Pengujian terhadap sebuah sistem secara umum bisa dilakukan dengan berbagai macam pendekatan, pada uji coba aplikasi ini akan dijabarkan dua buah metode pendekatan *testing* sistem, yaitu metode *white box* dan *black box*.

### 4.2.1 White Box Testing

*White Box Testing* yakni metode desain *test case* yang menggunakan struktur *control* desain prosedural untuk memperoleh *test case*. *Test case* dapat diperoleh dengan :

1. Menjamin bahwa semua *independent path* pada suatu modul telah digunakan minimal satu kali.
2. Menggunakan semua keputusan logis pada sisi *true* dan *false*
3. Mengeksekusi semua *loop* dalam batasannya dan pada batas operasionalnya.
4. Menggunakan struktur data internal untuk menjamin validitasnya.

Serangkaian tes ini dimaksudkan untuk meramalkan cara kerja perangkat lunak secara detail. Karenanya *logical path* (jalur logika) perangkat lunak akan dites dengan menyediakan *test case* yang akan mengerjakan kumpulan kondisi dan atau pengulangan secara fisik.

Selain berfungsi sebagaimana dijabarkan di atas, *white box testing* juga dilakukan mengetahui kesalahan-kesalahan yang tidak bisa dihandle oleh sistem (tidak ada validasi/pesan *error* dari program), ataupun keanehan-keanehan yang terjadi pada hasil (*output*) dari suatu hasil proses dalam program.

Kesalahan tersebut bisa disebabkan oleh kesalahan dalam logika program, *syntax*, atau *code* program, dimana kesalahan tersebut hanya *programmer* saja yang mengetahuinya. *User* hanya akan mengetahui *output* yang dihasilkan berbeda dengan yang diharapkan.

### 4.2.2 Black Box Testing

*Black box testing* merupakan tahap pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Memungkinkan pengembang perangkat lunak untuk mendapatkan serangkaian kondisi *input* yang mempergunakan semua persyaratan fungsional program. *Black box testing* bukan alternatif *white box testing*, namun merupakan pelengkap yang mampu mengungkap kesalahan, jika dibandingkan metode *white*

*box testing*, *black box testing* berusaha menemukan kesalahan dalam beberapa kategori yaitu :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data/akses *database* eksternal
4. Kesalahan kinerja, inisialisasi dan kesalahan terminasi.

## **4.3 Manual Program**

### **4.3.1 Cari Jalur**

Langkah awal *user* harus berada pada *form* cari jalur terlebih dahulu, Setelah itu *user* mengisi *textfield* berangkat dan tujuan. Untuk meminimalisir terjadinya kesalahan penulisan nama halte, *user* bisa memilih tombol OK pada *textfield* berangkat atau tujuan yang kosong, sehingga sistem akan menampilkan *list halte* secara keseluruhan. Kemudian *user* memilih salah satu nama halte sesuai pencarian dengan mengklik halte yang dimaksud pada *list halte*.

Jika sudah mengisi *textfield* berangkat dan tujuan *user* bisa memilih menu cari, aplikasi akan merespon dengan mencari data sesuai isi *textfield* berangkat dan tujuan, apabila data ada dan tidak terdapat kesalahan penulisan nama halte maka aplikasi akan menampilkan *list jalur* yang sesuai sedangkan apabila data tidak ada atau terjadi kesalahan penulisan halte maka aplikasi tidak akan menampilkan *list jalur*.

Langkah terakhir *user* dapat melihat rute yang dilalui oleh jalur tersebut dengan mengklik salah satunya.

### **4.3.2 Lihat Rute**

Pada sub bab sebelumnya di jelaskan *user* dapat melihat rute dengan cara mengklik nama jalur. Jadi saat berada pada *list jalur* *user* dapat mengklik salah satu nama jalur untuk melihat halte-halte yang dilalui oleh jalur tersebut secara urut dan keseluruhan.

Untuk melakukan aktifitas lihat rute tidak hanya dapat dilakukan pada saat *user* telah melakukan aktifitas cari jalur, melainkan dapat langsung membuka *list jalur* dari awal dengan cara memilih menu lihat rute yang pada halaman utama. *List jalur* nantinya akan menampilkan keseluruhan jalur yang dapat diklik salah satunya untuk melihat rute jalur tersebut.

#### 4.3.3 Lihat Info

Untuk melihat info *user* dapat melakukannya dengan cara memilih menu info pada pilihan menu di halaman utama.

#### 4.3.4 Cek Update

Untuk cek *update* dan mengunduh aplikasi versi terbarunya di kemudian hari disediakan pilihan menu cek *update* pada halaman utama.

### 5. Penutup

#### 5.1 Kesimpulan

Setelah beberapa tahapan dalam menyelesaikan aplikasi *mobile* untuk mengetahui jalur bus Trans Jogja, dapat disimpulkan bahwa :

1. Aplikasi *mobile* Trans Jogja dapat dipergunakan untuk menampilkan dan menginformasikan jalur, rute, halte, dan informasi singkat tentang bus Trans Jogja.
2. Aplikasi dapat diakses melalui telepon seluler yang sudah mendukung *Java* di dalamnya dan aplikasi dapat diakses di mana saja dengan cepat karena tidak membutuhkan koneksi internet pada saat pengoperasiannya.
3. Pembuatan aplikasi *mobile* ini menggunakan perangkat lunak Netbeans 6.5 dengan bahasa pemrograman *Java 2 Micro Edition*.

#### 5.2 Saran

Penulisan skripsi ini tentu terdapat banyak kekurangan yang mungkin dapat disempurnakan lagi pada penelitian-penelitian berikutnya. Maka agar aplikasi ini dapat lebih sempurna lagi terdapat beberapa saran yang mungkin dapat dipergunakan antara lain :

1. Adanya penambahan fitur agar aplikasi menjadi lebih kompleks dan pengembangan logika program untuk optimasi kerja sistem.
2. Sistem dapat dikembangkan menjadi lebih terstruktur dengan implementasi database, sehingga memudahkan *update* pada saat ada penambahan-penambahan jalur, rute, halte, maupun informasi Trans Jogja lainnya.
3. Tampilan pada aplikasi *mobile* ini masih sangat sederhana (berbasis teks), diharapkan dalam pengembangannya dapat memperindah tampilan dengan menambahkan gambar, animasi, ataupun suara agar lebih menarik.

4. Pembuatan aplikasi *mobile* ini masih bersifat sederhana dan percontohan, untuk itu diharapkan selanjutnya disempurnakan lagi agar dapat benar-benar diimplementasikan kepada para pengguna jasa bus Trans Jogja.

## 6. Daftar Pustaka

- Dishub DIY. 2010. Informasi Trans Jogja. [http://dishub-diy.net/index.php?option=com\\_content&view=category&id=40&Itemid=283](http://dishub-diy.net/index.php?option=com_content&view=category&id=40&Itemid=283). diakses 29 Maret 2010.
- Fowler, M. 2004. *UML Distilled*, edisi ketiga. Yogyakarta: Andi Offset.
- Jeni. 2007. Pengembangan Aplikasi *Mobile*. <http://poss.ipb.ac.id/files/JENI-J2ME-Bab01-Pengembangan%20Aplikasi%20Mobile.pdf>. diakses 7 April 2010.
- Jogiyanto, Hm. 2005. Analisis dan Desain Sistem Informasi Pendekatan Terstruktur, edisi ketiga. Yogyakarta: Andi Offset.
- Nugroho, Y. 2003. J2ME – MIDP. [http://kur2003.if.itb.ac.id/file/IF2281\\_J2ME-MIDP.pdf](http://kur2003.if.itb.ac.id/file/IF2281_J2ME-MIDP.pdf). diakses 3 Maret 2010
- Setiawan, T A. 2009. Rute Trayek Bus Trans Jogja. <http://transjogja.net/rute/>. diakses 15 Februari 2010.
- Tim Penyusun. 2009. Pedoman Penyusunan Proposal dan Laporan Skripsi. Yogyakarta: S1 Sistem Informasi & Teknik Informatika STMIK Amikom Yogyakarta.
- Umami, L. 2009. Modul *Java Micro Edition Mobile Application*. Yogyakarta: Jogja IT Training Center.